REMARKS

The Examiner has withdrawn th previous rejection based upon the Adelman et al. patent, and has imposed a new rejection based upon anticipation under 35 USC 102 by new prior art, Oki et al., U.S. 6,735,206 (hereafter just Oki).

More specifically, in sections 3 to 18 of the office action, claims 1-7, 12-24 adn 29-34 have been rejected as anticipated by Oki. It is believed by the undersigned that this is a mistake as there are only 27 claims in this case. The undersigned will treat all 27 claims as being rejected as anticipated per agreement with Examiner Ismail after a telephone conference regarding this issue.

Oki discloses a clustered comuting system which includes a set of served nodes as well as an interface node. The interface node distributes the service request from the client to the server nodes.

The interface node handles all service requests and distributes them to the cluster of server nodes.

To support a prima facie Section 102 rejection, an anticipating reference must contain every element of the rejected claim, united in the same way to perform the same function.

Oki fails to teach a network cluster having at least two nodes and each nod handling separate sets of data packets. In Oki, the interface node handles the same sets of data packets as the server nodes. Claim 1 calls in the preamble for each node to handle separate sets of packets, and then, in the main body of the claim details how the first node contains both node specific and common data structures so that the node can know which sets of packets it is handling and which sets of packets the other nodes are handling. This feature is not found in Oki.

Oki fails to teach maintaining in a first node a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in the first node. This feature is found in claim 1 in the following language:

> - maintaining in a first node a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in said first node,

In independent claims 14 and 23, this feature is found in the following claim language:

- means maintaining in said first data storage a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in said node,

Independent Claim 26 has been amended to put it into proper form for a computer product claim. As amended, claim 26 includes the following element not anticipated by Oki:

<u>-maintaining in a first node a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in said first node,</u>

Independent claim 27 has been amended to put it into proper form for a computer product claim stated in means plus function form. As amended, claim 27 includes the following element not anticipated by Oki:

<u>-means for maintaining in a first node a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in said first node,</u>

The Examiner refers to column 5, lines 42-64 in Oki as teaching this feature, and alleges that the interface node 103 maintains node-specific data structure needed for handling sets of data packets.

However, the interface node 13 is not a node which handles separate sets of data packets as any other node in the cluster does. On the contrary, the interface node forwards all service requests to the server nodes. Further, Oke does not teach any state information relating to sets of data packets. Col. 5, lines 46-64 discloses that the interface node stores information relating to load balancing policies and the configuration of the cluster network rather than state information needed for handling sets of data packets handled in the interface node.

Oki further fails to teach maintaining in the first node, in addition to the node-specific data structure, a second, common data structure. This common data structure is comprised of at least entries representing state information for data packets handled in at least one other node of the network element cluster. The contents of the common data structure effectively differ from the entries maintained in a node-specific data structure and includes copies of all state information entries maintained in a node-specific data structures of the other nodes in the cluster. In the language of the claims, the common data structure in the first node contains copies of the data in the node-specific data structure of said at least one other node in the cluster and which is needed for handling sets of data packets in said at least one other node.

The entries are maintained in the common data structure according to information on how different sets of data packets are distributed among the nodes of the network element cluster.

Claim 1 contains the following language which brings out this feature which distinguishes the claim from the teachings of Oki:

> - maintaining in said first node in addition to said node-specific data structure a second, common data structure comprising at least entries representing state information for data packets handled in at least one other node of said network element cluster, the contents of said common data structure effectively differing from the contents of said node-specific data structure and including copies of all state information entries maintained in a node-specific data structure of said at least one other node and needed for handling sets of data packets in said at least one other node, said entries being maintained according to information on how different sets of data packets are distributed among the nodes of the network element cluster,

The Examiner again refers to Col. 5, lines 42-64 in Oki. However, as noted above, the interface node only maintains information on a load distribution policy and the configuration of the network element cluster. There is no first, node-specific data structure comprising state information for handling sets of data packets handled in the interface node, and there is no teaching of a second common data structure comprising state information for data packets handled in at least one other node of the network element cluster. There is no teaching of first and second data structure including state information in the interface node. Further, there is no teaching that the contents of the second common data structure effectively differs from the contents of the first, node-specific data structure. Further, there is no teaching that the second common data structure includes copies of all state information entries maintained in a node-specific data structure of said at least one other node and needed for handling sets of data packets in said at least one other node. There is no teaching of any node-specific data structure containing state information and maintained in any of the server nodes. Further, Oki fails to teach that such entities would be maintained according to information on how different sets of data packets are distributed among the nodes of the network element cluster.

Oki further fails tc teach dynamically changing distribution of at least one set of data packets from said at least one other node to said first node of a network element cluster. To the opposite, in the sytem according to Oki, the interface node distributes services to server nodes, i.e., it provides the cluster networking multiplexing. In claim 1, the language which defines this feature is:

> - dynamically changing distribution of at least one set of data packets from said at least one other node to said first node the network element cluster, and providing

said first node with respective changed distribution information,

Further, Oki fails to teach that in response to re-distributing at least one set of data packets from at least one other node to the first node, the state information entries in the common data structure of the first node pertaining to the re-distributed set of data packets is transferred out of the common data structure of the first node into the node-specific data structure of the first node. The language of claim 1 that reflects this mode of operation of the invention is:

> - in response to said changed distribution information, selecting the state information entries of said at least one re-distributed set of data packets from said second common data structure and transferring them to said first node-specific data structure of said first node.

The Examiner refers to Col. 7, lines 34-55 as teaching these features. However, this section in Oki only teaches that the interface node uses a random or pseudo-random hash function to select a server node to forward a packet to. This is not teaching that the handling of at lest one set of data packets already handled by another node in the network element cluster, is transferred to the interface node to be handled by the interface node, or that the interface node would in response to such a change select all state information entries of said at least one re-distributed set of data packets from a second common data structure of the interface node and transfer these entries to a node-specific data structure of the interface node as called for by the claimed invention. , it is also not a teaching that a second common data structure in the interface node includes copies of all state information entries of the first node-specific data structure of said other node from which the data packets were transferred.

Thus, the subject matter of Oki is not anticipated by Oki et al. Similarly, independent claim 14 for a network element node and independent claim 23 for a network element cluster, as well as claims 26 and 27 for a computer program and a computer program product are not anticipated by Oki. Specifically, claim 14 contains the following language which distinguishes it from Oki:

> - means maintaining in said first data storage a first, node-specific data structure comprising entries representing state information needed for handling sets of data packets handled in said node,
>
> ....
>
> - means maintaining in said second data storage a second, common data structure comprising at least entries representing state information for data packets handled in one other node of said network element cluster, the contents of said common data structure effectively differing from the contents of said node-specific data structure and including copies of all state information entries maintained in a node-

specific data structure of said at least one other node and needed for handling sets of data packets in said at least one other node, and said entries being maintained according to information on how different sets of data packets are distributed among the nodes of the network element cluster,

- means receiving changed distribution information dynamically changing distribution of at least one set of data packets from said at least one other node to said node in the network element cluster, and

- means that, based on said changed distribution information selects the state information entries of said at least one re-distributed set of data packets from said second common data structure in said second data storage and transfers them to said first node-specific data structure in said first data storage of said node.

Similar language is found in claim claim 23. Claims 26 and 27, as amended, contain similar language from claim 1 which distinguishes over Oki for the reasons argued above.

In sections 19-20 of the office action, claim 1 was rejected as obvious over applicant's admitted prior art (AAPA) in view of Oki et al.

As AAPA, the Examiner refers to the description of prior art made in our own application. Applicant believes that the preamble of claim 1 (quoted below) is a feature known in the prior art as alleged by the Examiner on page 8 of the Final Rejection. That element is:

> A method for handling dynamic state information used for handling data packets, which arrive at a network element node of a network element cluster, **said network element cluster having at least two nodes and each node handling separate sets of data packets**,

(emphasis added)

This portion of the claim refers to a conventional stateful cluster of network nodes.

However, the applicant has not admitted that it has been previously known in the art to maintain in said first node in addition to said first node-specific data structure a second, common data structure. Those features of the claimed invention are recited in the elements in the main body of claim 1 following the preamble.

The Examiner refers to Figures 1A and 1B, and page 2, paragraph 18 as teaching this feature. It is unclear whether the Examiner is referring to AAPa (which has no paragraph 18 as the paragraphs are not numbered but the lines are) or to Oki which has no Figures 1A and 1B.

Assuming the Examiner is referring to AAPA, we respectfully disagree that the applicant's own comments regarding the prior art teach a first node-specific data structure and a second, common data structure in each node of a cluster. Applicant admits that clusters are known where each node in the cluster handles different sets of data packets.

But AAPA at page 2, lines 17-23 teach each node filtering all packets to decide which ones to keep and process or, alternatively, a separate network switch processes all packets and distributes the various packets assigned to the various nodes to those nodes. That is not what the invention is.

Applicant would like to point out to the Examiner that on page 5 of the patent application at bar in discussing the prior art, it is explicitly stated that in prior art clusters both the active data structures 11b, 12b and 13b, and the passive state data structures 11a, 12a and 13a maintained in user space of the cluster nodes have all identical contents. The contents include entries related to all sets of data packets and handled in the cluster at a given moment of time (see page 5, lines 4-6; lines 13-16; lines 27-31). Those passage teach that in the prior art clusters, active data structures are maintained in kernel space and a copy of these active data structures is maintained in user space for synchronization purposes. As the entries in the active data structures are added and cleared over time, those changes are updated to the user space data structures. It is the user space data structures which are communicated with other nodes in the cluster. When incoming information to the user space data structures arrives from other nodes, the changes reflected in that incoming data are used to change the user space data structure and are then pushed to the kernel space data structure. This means that both the kernel space and user space data structures of every node in the cluster maintain entries about every set of data packets handled by every node in the cluster. (page 5, lines 13-16).

This description of related prior art is not a teaching that, like the claimed invention, a first, node-specific data structure exists in each prior art node of a cluster which comprises all the entries representing only state information needed for handling sets of data packets handled by that node and does not include any entries representing state information for sets of data packets handled in any other node of the cluster. In fact, the description in applicant's specification of related prior art on page 5 teaches the exact opposite of the invention: that the kernel data structure of each node contains entries for sets of data packets handled by every node in the cluster. This prior art scheme is disadvantageous because the size of the required data structure can be quite large and the time to find a match in the data structure entries for a newly received data packet can be large. Those are some of the problems the claimed invention is trying to solve, and, as such, are not the invention. In fact those AAPA teachings teach away from the invention and so the AAPA teachings do not provide

suggestion to one skilled in the art to combine this AAPA technology with Oki.

The description of related prior art in the applicant's disclosure also fails to teach that the first node has a second, common data structure having contents which effectively differ from the contents of the first, node-specific data structure. In fact, the AAPA description of the related prior art teaches just the opposite. It teaches that the kernel space data structure and the user space data structures are actively maintained to be the same. In the claimed invention, the common data structure maintains copies of entries in the node-specific data structure of at least one other node in the cluster with the entries in the common data structure of each node in the cluster being maintained according to information on how different sets of data packets are distributed among the nodes of the network element cluster.

In contrast, the AAPA description of the applicant's specification teaches that in the prior art, all data structures discussed on page 5 and shown in Figure 1B of the applicant's drawings of the prior art have an identical content. Further, there is no mention of how the entries in these data structures are organized within the data structure. The Examiner alleges that the features claiemd in the last two elements of claim 1 are taught by Oki et al., and that it would have been obvious to combine the teachings of Oki et al. with AAPA in order to achieve the invention as claimed.

This combination of prior art teachings does not support a proper prima facie case of obviousness. First, as noted above, the first and second data structures according to claim 1 are not taught in the AAPA description of the related prior art in the applicant's specification. The claimed invention in claim 1 requires that the node specific and common data structures be different in content. The AAPA description requires that the kernel space and user space data structures be actively maintained to be identical. This teaches away from the claimed invention and does not support the suggestion needed to support the prima facie case of obviousness based upon combination of references.

Further, Figures 1A and 1B illustrate a network configuration wherein all nodes share a common IP address for addressing the cluster. All nodes see all data packets arriving at a cluster and are capable of distinguishing which data packets belong to each node. Each node processes only those packets that are assigned to it and ignores other data packets.

Oki teaches a special purpose interface node for distributing/multiplexing the incoming data packets. There would have been no motivation for a skilled person in the art to apply any teachings relating to such an interface/multiplexer node according to Oki in the cluster of the

AAPA shown in Figures 1A and 1B in the present invention. The reason no motivation for the combination exists is because the AAPA teaches the kernel and user space data structures to be identical in content and the invention requires that the node specific and common data structures be different. Thus, one skilled in the art would recognize that even in the combination of AAPA with Oki were to be made, the combination would not work to solve the problem of the invention. There is therefore no liklihood of success to support the notion that the suggestion needed for a prima facie obviousness case exists.

The Examiner further alleges that Oki teaches a method and apparatus taht uses a destination address to perform fast lookup to determine a service for a packet, and that Oki uses a packet distribution table (PDT) to implement load balancing among the nodes of the cluster.

Oki may indeed perform these functions, but they have nothing to do with the present invention. These functions relate to the operation of the interface node/cluster networking multiplexer for distributing packets. They relate to a network node which is not preent in the cluster configuration shown in Figures 1A and 1B in the present application. These functions do not relate to any state information maintained in the actual cluster nodes.

According to the Examiner's further arguments, Oki teaches that if a server fails, then another server will take over processing, and checkpointing process ensures that the configuration data will be present in the new server that was before available in the old server in order for the new server to process the packet. The Examiner refers to Col. 7, lines 34-55 of Oki. However, this section in Oki makes no such disclosure. It only describes the distribution of traffic during a normal operation.

The checkpointing process referred to in the next paragraph, lines 58-63 in Oki refers to maintaining redundant configuration data in a secondary PDT server. Once again, this relates to redundant operation of the interface node/multiplexer not present in the cluster configuration shown in Figures 1A and 1B.

There would have been no motivation for a person skilled in the art to apply teachings of Oki in the AAPA cluster system shown in Figures 1A and 1B of applicant's specification.

Further, even if a person skilld in the art had applied teachings of Oki in the AAPA network configuration of Figures 1A and 1B, he would have introduced a further element to the cluster, namely the interface node/multiplexer. In this scenario, the person skilled in the art would not have achieved the subject matter of claim 1 where the re-distribution of a set of

data packets to a first node from another node in the cluster causes the state information of the transferred set of data packets to be transferred from the second common state information structure of the first node into the first node-specific state information data structure within the first node.

Based upon the above arguments, claim 1 is not obvious over the AAPA described by the applicant in view of Oki.

Allowance of the existing claims is respectfully requested. A notice of appeal is filed herewith.

Dated: February 22, 2006

Respectfully submitted,

Ronald C. Fish

Ronald Craig Fish
Reg. No. 28,843
Tel 408 866 4777
FAX 408 866 4785
RCFIPLAW@COMCAST.NET
WWW.HITECH-PATENTS.COM

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail, postage prepaid, in an envelope addressed to: Commissioner for Patents, Mail Stop After Final, P.O. Box 1450, Alexandria, Va. 22313-1450. on 2/22/06

(Date of Deposit)

Ronald C. Fish

Signature of Depositer